Optimal Coordinate to Add New of Hubway Station

Riken Maharjan, Biken Maharjan

rikenm, bm181354{@bu.edu}

Boston University

CS591 L1 Data Mechanics, Spring 2018

Objective

The objective of the project is to find the optimal coordinates to add new Hubway Stations.

Introduction

Hubway is a bicycle sharing company where guest and subscribed users can lend a bike for certain amount of time. Our main goal was motivated by the question what if Hubway would like to add more station in the city of Boston. To obtained this objective, the clustering algorithm like hierarchal and k-mean were used to place the new bike stations.

Data.

The following publicly available dataset were used to help determine the most optimal coordinates to place the new stations in the greater Boston and city of Boston.

- [Hubway Trip data] https://s3.amazonaws.com/hubway-data/index.html
- [Hubway Station data] https://s3.amazonaws.com/hubway-data/Hubway_Stations_as_of_July_2017.csv

Methods

After retrieving the data, we basically, at first, did hierarchal clustering based on the distance. We decided 20 clusters seem appropriate. From the first dataset, we calculated how many times each station is visited in period of three month.
We stored this value in an array as popularity for each station. User would select how many stations to add by sliding the slider in our webpage.
We then decided to put station on the cluster with the highest average popularity.
After this we use k-mean to figure out the exact coordinates to put our stations.

The algorithm we created had three parts. They are as followed.

i) Hierarchal Clustering.

Hierarchical clustering is a clustering where we start with n clusters where n is the total points. To apply this algorithm, we need to first create n^2 matrix (*pdist*) where element $e_{(i,j)}$ is the distance between station *i* and station *j*. This distance can be Euclidian or any similarity or dissimilarity scores. We used haversine distance for our distance in our matrix.
Hierarchal clustering uses this matrix to combine two stations or clusters into a bigger cluster. The two stations or clusters with the smallest distance between them in matrix are combined first. In this way slowly and steadily, it uses matrix created by similarity equation like euclidian or haversine distance to combine clusters which have the minimum distance.

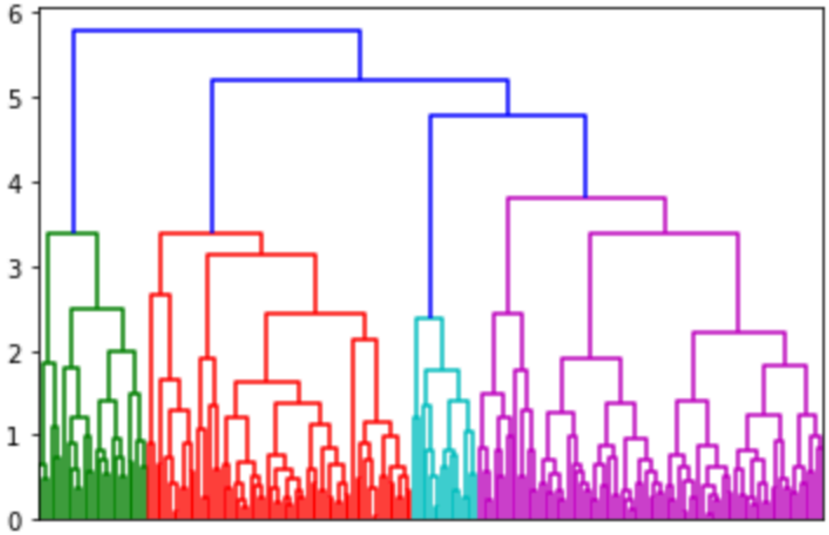

fig 1.0 Dendrogram of Hierarchical clustering

The figure above shows how the algorithm choices points to accumulate. Slowly the number of cluster decreases and at the end we would have one cluster. The algorithm was stopped before this point. 20 clusters seem decent point to stop.

ii) Calculating Average popularity

We would calculate average popularity of *n* clusters. Our web app would ask how many stations you would like to add. User would give a number by dragging a slider. We would see iterating over average popularity which cluster needs a station. As we keep on adding stations in cluster based on average popularity, the average popularity of the cluster would go down. We would add station in a cluster until another cluster has greater average popularity.

iii) Weighted K-mean

After we have decided how many stations to be added per cluster from step ii) we would now do weighted k-mean algorithm per cluster. Some of the cluster might not get any stations to be added. If cluster A has 2 station to be added then we would do weighted k-mean with two means in cluster A. The way we weighted the k-mean is by duplicating points in a cluster by their popularity. If station x belongs in cluster A and station x has 200 popularity then we would duplicate x 200 times when doing K-mean in cluster A.

Visualization

D3 and web services, which was created from flask/python were used. We stored all our result from our algorithm into Mongodb. This result will be obtained by our web service which sends this data to front end in D3 from our Flask app when user slides the slider. When ever user moves the slider, we are just fetching data from our mongodb with a get request to our web service.

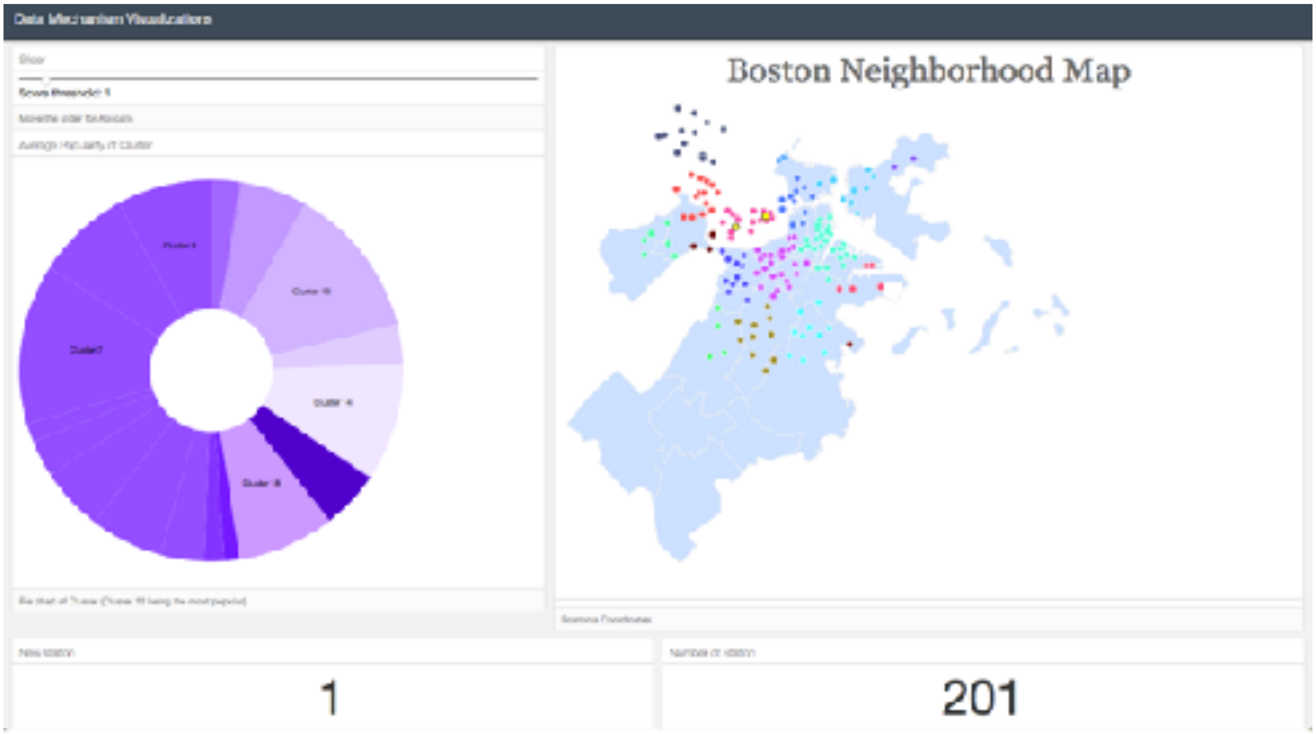

Fig 2.0 Visualization

Moving the slider basically adds stations with yellow color in the cluster. The air chart shows the average popularity after addition of new station. The decrement is very small with each addition.

Results

Our algorithm showed that Cambridge would have the most amount of bicycle stations added as clusters over there had the higher average popularity. This makes sense as Cambridge has many student bicycle rider. If Hubway was to add new station, it should be firstly in the Cambridge area.

Future Work

Moving forward, there are many things that we can improve on, some of which are:

Expand datasets to include other factors like income, population distribution to score instead go just using average popularity of the cluster.